

A New Mechanism for Maintaining Diversity of Pareto Archive in Multiobjective Optimization

Jaroslav Hájek¹, András Szöllös¹, Jakub Šístek^{2†}

¹ Aeronautical Research and Test Institute,
Beranových 130, Praha, CZ-199 05, Czech Republic

² Institute of Mathematics of the AS CR,
Žitná 25, Praha, CZ-115 67, Czech Republic

†Corresponding author

hajek@vzlu.cz, szollos@vzlu.cz, sistek@math.cas.cz

Abstract

The article introduces a new mechanism for selecting individuals to a Pareto archive. It was combined with a micro-genetic algorithm and tested on several problems. The ability of this approach to produce individuals uniformly distributed along the Pareto set without negative impact on convergence is demonstrated on presented results. The new concept was confronted with NSGA-II, SPEA2, and IBEA algorithms from the PISA package. Another studied effect is the size of population versus number of generations for small populations.

Keywords: multi-objective optimization; micro-genetic algorithms; diversity preserving; Pareto archive; selection to archive

1 Introduction

The field of multiobjective design optimization has evolved very fast during last years, reflecting the need of solving tasks with several conflicting criteria, which is common in practical problems. From the mathematical point of view, this corresponds to minimization/maximization of a vector-valued function, which rarely leads to a single solution. Consequently, a whole hyperplane of trade-off solutions, called Pareto-optimal set, is expected as the result instead of a single optimum.

A number of algorithms have been presented that generate a set of solutions approximating this hyperplane. The quality of the approximation is usually considered from two points of view: (i) the closeness to the exact trade-off surface and (ii) its distribution. The former is related to convergence properties of an algorithm while the latter describes its ability to maintain diversity. An ideal algorithm should produce well converged solutions perfectly distributed along the Pareto front. However, these requirements are conflicting, and many current approaches concentrates on one of them finding reasonable compromise in the other.

In this study, our attention is focused on the second aspect of diversity of the Pareto-optimal set, namely we present a new strategy for maintaining variety of members of a Pareto archive.

The problem of maintaining uniform distribution at an affordable cost has been addressed by many algorithms. It is known that the notion of crowding distance proposed by Deb et al. for algorithm NSGA-II [5, 6] is not sufficient to maintain diversity of the evolution for more than two objectives (e.g. [7, 8]). On the other hand, SPEA2 by Zitzler et al. [18] is usually able to produce well spread solutions even for three or more objectives. The concept of archiving promising design vectors was first introduced for SPEA by Zitzler and Thiele

[19]. Knowles and Corne presented the Pareto Archived Evolutionary Strategy (PAES) [11] and proposed the adaptive grid algorithm ([12]) to maintain diversity. However, it is difficult to keep the efficiency of this approach in cases with more than three objectives.

The new mechanism presented in this paper was implemented in micro-genetic algorithm μ ARMOGA proposed by Szöllös et al. [15], and results for three standard three-objective benchmark problems are presented.

Our second aim is to investigate the effect of population size for small (sometimes called micro) populations on the performance of μ ARMOGA. It was reported by Krishnakumar [13] for single-objective optimization and by Coello and Pulido [3] for multiple objectives, that very small populations can lead to fast convergence to the Pareto front. In this context, most experiments were performed using populations of 4, 10 and 20 individuals.

Results got by μ ARMOGA equipped with the new archiving mechanism are compared with those obtained by two leading methods in the field, namely NSGA-II by Deb et al. [6] and SPEA2 by Zitzler et al. [18], and a recent interesting algorithm IBEA by Zitzler and Künzli [17]. All these are implemented in the Platform and Programming Language Independent Interface for Search Algorithms (PISA)¹ [2]. PISA is an interesting open source package developed by the team of Prof. E. Zitzler at ETH Zürich. The software implements various selection, crossover, and mutation operators and objective function evaluations. An important idea of the project is to separate the selection of promising candidates from objective function evaluation, crossover and mutation and implement these in two separate programs, interchanging information via formatted files. These programs are called *selectors* and *variators* in the PISA context. There is an increasing number of ready-to-use variators and selectors that can be downloaded from the web page of the PISA project. Therefore, the system offers

¹<http://www.tik.ee.ethz.ch/sop/pisa>

a simple way to produce fair comparisons of various selection schemes with the same variator. While the described scheme of splitting an evolutionary algorithm into two separate programs is very useful for some techniques, in our opinion, it does not fit to algorithms with strong coupling between both stages via the use of an archiving procedure. That is the reason why our implementation of μ ARMOGA was used, instead of integrating the proposed archiving technique into the PISA framework.

Three metrics, measuring both convergence to the exact front and diversity of the approximate set, are used for the comparison. It is observed, that the new algorithm produces very good distribution of individuals outperforming in this respect the other algorithms in many cases. The archiving strategy does not seem to affect its convergence. Moreover, diversity is maintained in an affordable way as suggested by presented numerical experiments.

The rest of the paper is organised as follows. In Section 2, μ ARMOGA is recalled with an emphasis on its main aspects. Section 3 contains the main contribution, which is the proposition of a new archiving mechanism. Tests and comparisons with the other evolutionary techniques can be found in Section 4, where we describe the test problems (4.1), metrics used for evaluating the performance (4.2), detailed setting of particular algorithms (4.3), and organization of the experiments (4.4), respectively. Our findings are discussed in detail in Section 5, while Section 6 contains summary of the work and concluding remarks.

2 μ ARMOGA – Multiobjective micro-genetic algorithm with range adaptation: A Brief Introduction

To minimize the costly evaluation of individuals, it is straightforward to see that one way to go is to minimize their number. It is well known for evolutionary practitioners, that using smaller populations and applying the evolutionary operators many times is often more favourable than vice versa (e.g. [14]). This idea can be brought to an extreme by using a micro-population (e.g. 4, 5, 10 individuals), what we really did when we utilized some ideas of Krishnakumar [13] and of Coello and Pulido [3]. Krishnakumar came with the concept of micro-genetic approach first, and used it for single-objective optimization. His algorithm contained only selection and crossover operators, and no mutation operator. Instead, the author introduced a reinitialization technique, which was invoked once in a few generations to ensure diversity for the evolution. The latter two researchers proposed a micro-genetic algorithm enabling to tackle multi-objective problems. Their concept was similar to that of Krishnakumar, i.e. it contained selection, crossover, and reinitialization operators supplemented by a mutation operator. Both algorithms were verified on various test problems. In both the cases, the micro-genetic variants converged to the optimum (Pareto-front) much faster than their macrogenetic counterparts used for comparison.

In the approach by Szöllös [15], microgenetic algorithm is supplemented by range adaptation and “knowledge-based” reinitialization procedure exploiting the Pareto-archive to generate better individuals.

The concept of range adaptation was originally introduced by Arakawa and Hagiwara [1], who used it with binary coding of the design variables. Its essence lies in ability to promote the evolution towards promising regions of the design

space via sophisticated manipulation with the population statistics. Due to the coding, it contained some artificial parameters which were hard to guess in general. Oyama [14] used range adaptation in real domain and was successful in avoiding this drawback via encoding a design variable to a real number $r_i \in (0, 1)$ defined by integration of the Gaussian distribution $N(0, 1)$

$$r_i = \int_{-\infty}^{\tilde{p}_i} N(0, 1)(z)dz, \quad (1)$$

where \tilde{p}_i is linked to the original design variable p_i by

$$p_i = \sigma_i \cdot \tilde{p}_i + \mu_i. \quad (2)$$

We are using this encoding scheme too, with one important difference: Oyama originally calculated the average μ_i and the standard deviation σ_i by sampling the upper half of the population, which is justified as long as macro-populations are used (e.g. with more than fifty individuals). But such approach would be too restrictive in the case of microevolution, since the upper half of the micro-population contains too little information to keep the diversity. Consequently, the evolution quickly ends up in premature convergence. Thus, we calculate both by taking into account the whole population.

“Knowledge based” reinitialization resulted from an attempt to use the members of the Pareto-archive to get new members, superseding the old ones by putting several of them into the reinitialized population. Moreover, only a subset of the archive is considered. For instance, two archive members with extreme values of two different objectives chosen randomly are usually exploited. In this way, it is possible to further improve the whole archive by improving its subsets. The functioning of μ ARMOGA can be seen in Figure 15. After initialization of the population by Latin hypercube sampling (LHS) and evaluation depicted as

archive update, the evolution goes through selection, mating and mutation to evaluation of the new population. Each n -th generation the population statistics is updated, range-adaptation takes place, followed by knowledge based (elitist-random) reinitialization. A thorough description of the algorithm is to be found in [15].

Our approach contains two new system parameters: adaptation factor δ and minimal standard deviation σ_{min} . In short, μ ARMOGA strives to keep the evolution in a permanently “excited” state via forced modification of the population statistics. It practically means that the standard deviation is not allowed to fall under certain minimal value of σ_{min} for any design variable. This helps to prevent the micro-genetic algorithm from getting stuck in premature convergence. The role of the adaptation factor δ lies in controlling the frequency of range adaptation: if reinitialization is necessary, and the new standard deviation of a design variable is changed by more than $\delta \cdot \sigma_{old}$, where σ_{old} is the standard deviation when the last reinitialization took place, then the range of that design variable is adapted.

3 The archiving algorithm

Pareto archive is a key component of many evolutionary algorithms. It acts as a collector of good individuals during the evolution, and is often used to give the resulting Pareto front approximation at the end of the evolution. After new individuals are evaluated, the archive is improved if these individuals dominate or are non-dominated with respect to the existing individuals of the archive. During reinitialization, the micro-genetic algorithm retrieves information from the archive, using it to explore the promising regions of the search space.

Obviously, in any real setup we must limit the number of individuals stored in an archive. This is necessary not just to keep the amount of information pro-

cessed feasible, but also to get a good diversity of the resulting approximation. In our strategy, we use a fixed upper limit on the number of individuals stored in the archive. Ideally, we want to end up with a full archive of Pareto-optimal solutions that is “well spread” over the true Pareto front of the problem. Our approach is an archive dealing with a single new individual at a time. This is particularly suitable for micro-evolutionary approaches, where we only have a few new individuals from each generation.

When a new individual arrives, it is first checked for Pareto dominance with all existing members of the archive. Now, we distinguish among three cases:

- The new individual is dominated by one or more members of the archive. In this case, the new individual is discarded.
- The new individual dominates one or more members of the archive. The dominated ones are removed, and the new individual is added to the archive and the internal information of the archive is updated (see below).
- The new individual is non-dominated and non-dominating. If the number of members of the archive has not yet reached the upper limit, the new individual is added as in the previous case. In the opposite case, we need to discard at least one individual (either the newcomer or one from the archive), but we can not decide this by Pareto dominance. In this case, we proceed to the secondary decision procedure described below:

If we arrive at the case that can not be resolved by Pareto dominance, our secondary goal is to maximize the distance between neighbouring individuals, based on some distance-measure in the objective space. In this paper, we use the standard Euclidean distance, which is meaningful for any dimension of the objective space.

First, we consider the minimum pairwise distance, i.e.,

$$\min_{\substack{i,j \in P \\ i \neq j}} \|\mathbf{f}_i - \mathbf{f}_j\|, \quad (3)$$

where P denotes the set of archived individuals and \mathbf{f}_i stands for the vector of objective values of individual i . We take the pair of individuals that achieves the minimum in the above expression. If there are multiple pairs, we take any of them. Without the loss of generality, we assume that the minimum pair is $\mathbf{f}_1, \mathbf{f}_2$. Further, we denote the vector of objective values of the new individual as \mathbf{h} . If

$$\min_{\substack{k \in P \\ k \neq 1}} \|\mathbf{f}_k - \mathbf{h}\| > \|\mathbf{f}_1 - \mathbf{f}_2\|, \quad (4)$$

we can replace \mathbf{f}_1 by \mathbf{h} . Alternatively, if

$$\min_{\substack{k \in P \\ k \neq 2}} \|\mathbf{f}_k - \mathbf{h}\| > \|\mathbf{f}_1 - \mathbf{f}_2\|, \quad (5)$$

we can replace \mathbf{f}_2 by \mathbf{h} . If either of the above conditions is satisfied, the overall minimum pairwise distance will be improved by the substitution or, if there were multiple minimal pairs, it will stay the same but the number of minimal pairs will reduce. We call this as the *global improvement* check.

If neither of these conditions is satisfied, we consider the closest archived individual to \mathbf{h} , say, \mathbf{f}_c instead. If

$$\min_{\substack{k \in P \\ k \neq c}} \|\mathbf{f}_k - \mathbf{h}\| > \min_{\substack{k \in P \\ k \neq c}} \|\mathbf{f}_k - \mathbf{f}_c\|, \quad (6)$$

we replace \mathbf{f}_c by \mathbf{h} . If this condition holds, there is a certain subset of the archived individuals whose pairwise minimum will improve. This is the *local improvement* check.

If neither check is successful, we discard the new individual.

Searching for the minimum-distance pair of the archive afresh each time an individual is considered would be too costly. To make the procedure efficient, we maintain for each archived individual a pointer to its closest neighbour (or any of them). Therefore, searching for the pairwise minimum in (3) requires only one pass through the archive. Similarly, the right-hand side of equation (6) is simply the distance of \mathbf{f}_c to its closest neighbour. Hence, these two checks only require computing the distances of the new individual to all archived individuals, and computing the minima on left-hand sides of the equations (4), (5), and (6). Thus, *deciding* whether to add a new individual has linear complexity in terms of number of archived individuals (evaluating mutual pairwise dominance also has linear complexity).

If the new individual is to be added, the existing closest-neighbour links need to be updated. Each resulting archive member is considered in turn. If the link is valid (i.e. the closest neighbour in the archive was not discarded), we simply check if the newcomer is closer, and possibly update the link. This takes only constant time. However, if the link became invalid (the former closest neighbour was discarded), we need to compute the closest neighbour afresh by computing objective distances of the updated individual to all others.

It can be proven by a simple argument based on k -dimensional ball volumes that the maximum number of points in k -dimensional space having a single common closest neighbour is bounded from above by a constant depending on k . Since the Pareto archive consists of mutually non-dominating vectors, which can not be arranged arbitrarily, in our case the constant is even smaller. For instance, for a two-objective optimization, i.e. $k = 2$, a single archive member can be the closest neighbour to at most two other members at the same time.

Using this argument, it can be easily seen that the complexity of a single

archive update has complexity $O(N + MN)$, where N is the size of the Pareto archive and M is the number of archive members dominated by the new individual. As was already said, merely *deciding* whether the newcomer is to be added costs $O(N)$. If the decision is positive, there are two cases: either the newcomer dominates some M existing archive members, or it was added based on the secondary decision procedure. In the former case, M members will be discarded, so at most cM nearest-neighbour links will need to be updated, c being the upper bound constant discussed in the previous paragraph. In the latter case, one existing member is discarded, so at most c existing links must be updated. Given that updating a single link costs $O(N)$, together we have the cost $O(N + c(1 + M)N)$ which can be simplified to $O(N + MN)$, given that c is a constant independent of M, N .

While in principle M can be as high as N , in practice it drops to $M \ll N$ very quickly as the convergence proceeds and new dominating individuals become increasingly rare. It should also be noted that if M is high at one step, the evolution continues with an archive of $N - M$ which will be significantly smaller than N . Numerical experiments confirm that in real evolutionary runs, the average number of invalid links per archive update is very small, even much smaller than the theoretical bounds suggested above. This might be observed from Tables 19 and 20. Therefore, we can conclude that the procedure of adding new individual to our archive is essentially of linear complexity.

4 Comparison of results

The abilities of the new archiving mechanism are first demonstrated on test functions DTLZ1, DTLZ2 and DTLZ4, suggested by Deb et al. [8]. To examine the influence of population size, our algorithm was run separately with 4, 10 and 20 individuals. Obviously, it is preferable to maintain the number of

function evaluations as low as possible, therefore we study the behaviour of the aforementioned approaches for three fixed numbers of evaluations, 4 000, 20 000 and 40 000. For test problem DTLZ1, the number of function evaluations is extended to 100 000 and 200 000, since the algorithms were unable to converge to the global Pareto front with just 40 000 computations.

To further investigate the behaviour of the proposed method, we performed an experiment with test problem WFG1 suggested by Huband et al. [10]. For this difficult problem, it was necessary to run the evolution to as many as 2 000 000 evaluations to obtain reasonable convergence to the Pareto front.

4.1 Test problems

The algorithms are compared on three benchmark problems introduced in [8]. The following form of them is considered:

- DTLZ1

Minimize f_1, f_2, f_3 , where

$$f_1(\mathbf{x}) = \frac{1}{2}x_1x_2(1 + g(\mathbf{x}_M)), \quad (7)$$

$$f_2(\mathbf{x}) = \frac{1}{2}x_1(1 - x_2)(1 + g(\mathbf{x}_M)), \quad (8)$$

$$f_3(\mathbf{x}) = \frac{1}{2}(1 - x_1)(1 + g(\mathbf{x}_M)), \quad (9)$$

$$g(\mathbf{x}_M) = 100 \left(5 + \sum_{x_i \in \mathbf{x}_M} ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))) \right), \quad (10)$$

subject to $0 \leq x_i \leq 1$, for $i = 1, 2, \dots, 7$.

Here $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and $\mathbf{x}_M = (x_3, x_4, x_5, x_6, x_7)$.

- DTLZ2

Minimize f_1, f_2, f_3 , where

$$f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1\pi/2) \cos(x_2\pi/2), \quad (11)$$

$$f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1\pi/2) \sin(x_2\pi/2), \quad (12)$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1\pi/2), \quad (13)$$

$$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \quad (14)$$

subject to $0 \leq x_i \leq 1$, for $i = 1, 2, \dots, 12$.

Here $\mathbf{x} = (x_1, x_2, \dots, x_{12})$ and $\mathbf{x}_M = (x_3, x_4, \dots, x_{12})$.

- DTLZ4

Minimize f_1, f_2, f_3 , where

$$f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^{100}\pi/2) \cos(x_2^{100}\pi/2), \quad (15)$$

$$f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^{100}\pi/2) \sin(x_2^{100}\pi/2), \quad (16)$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1^{100}\pi/2), \quad (17)$$

$$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \quad (18)$$

subject to $0 \leq x_i \leq 1$, for $i = 1, 2, \dots, 12$.

Here $\mathbf{x} = (x_1, x_2, \dots, x_{12})$ and $\mathbf{x}_M = (x_3, x_4, \dots, x_{12})$.

Problem WFG1 is a benchmark problems introduced in [10]. The following form is considered:

- WFG1

Minimize f_1, f_2, f_3 , where

$$f_1(\mathbf{x}) = 2[(1 - \cos(z_1\pi/2))(1 - \cos(z_2\pi/2))], \quad (19)$$

$$f_2(\mathbf{x}) = 4[(1 - \cos(z_1\pi/2))(1 - \sin(z_2\pi/2))], \quad (20)$$

$$f_3(\mathbf{x}) = 6\left[1 - z_1 - \frac{\cos(10\pi z_1 + \pi/2)}{10\pi}\right], \quad (21)$$

where $z_1 = z_1(\mathbf{x})$, and $z_2 = z_2(\mathbf{x})$ are auxiliary variables obtained from design variables $\mathbf{x} = (x_1, x_2, \dots, x_{24})$ by a series of nonlinear transformations (see [10] for their definition). However, there is a slight difference in our application of transformation `b.poly`, which we use with exponent 0.2 instead of 0.02 suggested in [10] due to numerical issues in floating point arithmetic. Design variables have range $0 \leq x_i \leq 2i$, for $i = 1, 2, \dots, 24$.

The exact front of this problem is visualized in Figure 16.

4.2 Metrics

The results are evaluated according to three measures. The distance of members of the Pareto archive to the true Pareto front is measured using the **generational distance (GD)** [16], which is defined as

$$\text{GD} = \sqrt{\frac{1}{n} \sum_{i=1}^n d_i^2}, \quad (22)$$

where n is the number of nondominated solutions found by an algorithm, and d_i is the Euclidean distance of the i -th solution to the exact front. In order to evaluate the distance accurately, we implemented an approach, that is able to iteratively find the closest point of the exact front for each approximate solution, provided the analytic expression of the exact front is known. This point is then used for measuring the distance. Zero value of GD corresponds to all members

of the archive on the exact front.

We evaluate also another measure of convergence, denoted as **TOL5**. It is defined as the lowest value, such that $d_i > \text{TOL5}$ holds for at most 5 % of individuals. In statistics, it is called the 95-th percentile with respect to distance. Again, the lower value of TOL5 the better convergence. Zero value indicates, that at least 95% of archive members are on the exact front. This metric is less sensitive to remote individuals than the GD value.

The uniformity of distribution of archive members is measured by **spacing** defined in [4]. It is based on the distance to the nearest neighbour for each member of the archive, which is defined as

$$dn_i = \min_{\substack{j \in P \\ j \neq i}} \|\mathbf{f}_i - \mathbf{f}_j\|. \quad (23)$$

Now spacing is the ratio of standard deviation of values of these squared distances and their average, i.e.

$$\text{spacing} = \frac{1}{\overline{dn}} \sqrt{\frac{1}{n-1} \sum_{i=1}^n (dn_i - \overline{dn})^2}, \quad (24)$$

where \overline{dn} stands for the mean value

$$\overline{dn} = \frac{1}{n} \sum_{i=1}^n dn_i. \quad (25)$$

Consequently, zero spacing corresponds to uniform distribution of distances to the nearest neighbour. Although this does not assure global uniformity of distribution (e.g. for pairs of individuals), our experience with this metric is satisfactory.

The coverage of the Pareto front is not evaluated by means of a metric, but rather compared qualitatively at presented plots.

4.3 Setting of algorithms

All the algorithms from PISA package [2] (in the PISA context called selectors), i.e. NSGA-II, SPEA2, and IBEA, are used with the following setting of variator DTLZ:

- individual mutation probability ... 1,
- individual recombination probability ... 1,
- variable mutation probability ... 0.01 ,
- variable swap probability ... 0.5,
- variable recombination probability ... 1,
- η mutation ... 20,
- η recombination ... 15,
- use symmetric recombination ... 1,

For variator WFG, these values are the same except the value of variable mutation probability preset to 1 (default).

The simulations with PISA are performed with population of 100 individuals. All of them are selected for mating, producing 100 new individuals in each generation. The tournament of 2 individuals is used in these selectors. Experiments with IBEA are performed using the additive ϵ -indicator with scaling factor κ equal to 0.05.

The μ ARMOGA is run with 4, 10 and 20 individuals in population, marked as μ ARMOGA(4), μ ARMOGA(10) and μ ARMOGA(20), respectively (for WFG1, only 4 members of population are considered). The archive size is always set to 100 to produce results comparable with those of PISA algorithms. Simple one-point crossover scheme without mutation is used. It was reported by Oyama

[14], that this scheme derived for binary coded algorithms [9] works reasonably well also for real-domain. The version with 4 members uses reinitialization in each generation (DTLZ1, DTLZ2, DTLZ4) or once in four generations (WFG1), while for larger populations, the reinitialization is performed once per 3 generations for all problems. After reinitialization, several existing archive members are put into the new population. Their number is 2, 4 and 6 for the population of 4, 10 and 20 members, respectively. Random selection of individuals for mating is then performed with this modified population. The other important parameters of μ ARMOGA are set to the following values:

- adaptation factor $\delta \dots 1.4$,
- minimal standard deviation $\sigma_{min} \dots 0.8$ (DTLZ1, WFG1), 0.005 (DTLZ2, DTLZ4),
- recombination probability $\dots 1$.

Larger value of σ_{min} helps to attain the global Pareto optimal front of multimodal problems such as DTLZ1 and leads to faster convergence also in the case of WFG1. In general, its large values emphasizes global exploration of the design space while small values lead to refined search.

4.4 Experiments

The results for problems DTLZ1, DTLZ2 and DTLZ4 are summarized in Tables 1–11, and visualized in Figures 1–11. For problem WFG1, results are summarized in Tables 12–18 and Figures 12–14. The values in tables are obtained as averages for 20 different seeds and where it makes sense, the best value is emphasized by bold font. The approximation with the best distribution is selected out of the twenty runs of each algorithm for visualization. The exact Pareto front is marked by grid of small dots in presented figures. However,

not all twenty seeds lead to a successful approximation of the Pareto set in some instances, especially for problem DTLZ4. Most of the algorithms suffer from problems with robustness with respect to initial population and produce *degenerated* fronts for some seeds. We consider a front as degenerated, if all individuals have almost identical value of an objective, and thus cover just a line on the three dimensional surface of the exact front. Numbers of degenerated fronts for all problems and methods are summarized in Table 21.

The efficiency of the proposed archiving technique is further demonstrated in comparison with the same approach, i.e. μ ARMOGA, using crowding distance [5, 6]. That algorithm was described in [15]. Outputs of these experiments are summarized for DTLZ1 in Tables 22 and 23, for DTLZ2 in Tables 24 and 25, for DTLZ4 in Tables 26 and 27, and for WFG1 in Tables 28 and 29. Obtained Pareto fronts are plotted in Figures 17–28.

5 Discussion of results

5.1 DTLZ1

This problem with three objectives has a linear Pareto optimal front that intersects the axes of the objective space at value 0.5. Apart of the exact front, there exist a number of other parallel planes corresponding to local Pareto fronts. As these also attract an evolution, problem DTLZ1 tests the ability of a genetic algorithm to cope with multi-modality.

As can be seen in Figure 1, none of the algorithms is able to reach the global Pareto front in 4000 evaluations for any seed, and metrics in Table 1 do not provide much valuable information. However, we can remark that IBEA and μ ARMOGA(4) provide one order better convergence than the other algorithms and μ ARMOGA for all sizes of population provides reasonable spacing.

However, all algorithms except NSGA-II are able to reach the global front for some seeds in 20 000 evaluations (Figure 2). Comparing visually the results in Figure 2, we can conclude that μ ARMOGA(4) performs best, which is supported by the best values of all three metrics in Table 2. As individuals for many of the seeds are still away from the global front for all algorithms, metrics in Table 2 do not provide a detailed insight either.

The situation is further improved with 40 000 evaluations, for which all algorithms except NSGA-II are able to reach the global front for most of the seeds (Figure 3). However, Figure 3 shows that μ ARMOGA (for all sizes of population) produces the best distribution, which is confirmed by values of spacing in Table 3. Since for some seeds the individuals still are not in vicinity of the true Pareto front, the averaged metrics in Table 3 are still rather bad. According to Table 3, the best convergence is in average attained by μ ARMOGA(4) for this case.

For 100 000 and 200 000 evaluations, μ ARMOGA(4) achieves the global Pareto-optimal front for all seeds. All the other algorithms fail to find the global front for some seeds, which considerably spoils the metrics in Tables 4 and 5. Since IBEA produces only degenerated fronts in these cases, metrics are not evaluated and are omitted in Tables 4 and 5.

Although the distribution of fronts obtained by μ ARMOGA for all sizes of population is comparable to SPEA according to Figures 4 and 5, the metrics in Tables 4 and 5 reveal that spacing is, in average, one order better by μ ARMOGA than by SPEA. The best average convergence metrics are obtained by μ ARMOGA(4) (Tables 4 and 5).

5.2 DTLZ2

Problem DTLZ2 has three objectives, and the exact front corresponds to the part of a unit sphere when restricted to the octant given by non-negative values of all three objectives. This is the easiest problem for all compared algorithms and tests mainly the speed at which an algorithm is converging to the exact Pareto front.

Already for 4 000 evaluations, the fronts obtained by all the compared methods are reasonably converged and distributed along the exact Pareto front. Figure 6 shows that μ ARMOGA (regardless of the size of population) and SPEA produce the best distribution of individuals along the exact front, whereas the distribution obtained by IBEA is rather poor. This observation is confirmed by the spacing metric in Table 6. The best convergence is achieved by IBEA according to GD and TOL5 metrics in Table 6 followed by μ ARMOGA.

Similar observations can be made from the results for 20 000 evaluations (Table 7 and Figure 7) and 40 000 evaluations (Table 8 and Figure 8) – the best spacing is obtained for all sizes of population by μ ARMOGA and the best convergence is attained by IBEA, although the distribution of individuals along the Pareto front is worse.

5.3 DTLZ4

Although the definition of problem DTLZ4 is similar to DTLZ2 (cf. Section 4.1), the evolution is greatly influenced by the exponential transformation of design variables, which maps most of the space towards the axes in design space. This in turn pushes the evolution to the limits of the objective space. Thus, problem DTLZ4 tests best of the three DTLZ problems the ability of a genetic algorithm to obtain uniform distribution of individuals along the Pareto optimal surface.

For 4 000 evaluations, the best distribution is produced by SPEA2 (Figure 9).

This is confirmed by results of spacing in Table 9. However, μ ARMOGA(4) produces the best converged results.

For 20 000 evaluations, the distribution obtained by μ ARMOGA is already visually comparable with SPEA2 in Figure 10. Also spacing obtained by μ ARMOGA is comparable to that of SPEA2 according to Table 10 for 4 and 10 members of population. Algorithms μ ARMOGA(20), IBEA, and NSGA-II produce in average only slightly worse converged results. The best GD and TOL5 values are attained by μ ARMOGA(20).

In the case of 40 000 evaluations, the best distribution of individuals is attained by μ ARMOGA followed by SPEA2 according to Figure 11 and also confirmed by values of spacing in Table 11. The best convergence is again obtained by μ ARMOGA(20), followed by μ ARMOGA(10), μ ARMOGA(4) and IBEA, respectively (Table 11).

5.4 WFG1

This is a difficult problem and all tested algorithms had problems with convergence to the Pareto front. For this reason, number of evaluations of the objective function was increased to 2 000 000, after which some algorithms were able to attain the exact front.

After 4 000 evaluations, all algorithms produce results rather far from the Pareto optimal set (Table 12). Nevertheless, SPEA2 produces the most uniform distribution according to the spacing metric.

After 20 000 evaluations, μ ARMOGA(4) slightly leads in convergence followed by IBEA (Table 13), producing distribution with uniformity between SPEA2 (best spacing) and the rest of the algorithms. The same observations remain valid for 40 000 evaluations (Table 14).

After 100 000 as well as 200 000 evaluations, μ ARMOGA(4) dominates in

convergence to the exact front (GD and TOL5 metrics in Tables 15 and 16), producing distribution comparable with SPEA2 (best spacing). However, Figure 12 suggest, that $\mu\text{ARMOGA}(4)$ covers the whole Pareto front, unlike SPEA2.

Letting the evolution run to 1 000 000 and 2 000 000 evaluations, $\mu\text{ARMOGA}(4)$ dominates both in convergence (one order of magnitude compared to the second IBEA in Tables 15 and 16) and distribution along the exact Pareto front. In spacing metric, $\mu\text{ARMOGA}(4)$ is followed by SPEA2. Figures 13 and 14 show that distribution of individuals by $\mu\text{ARMOGA}(4)$ uniformly covers the whole Pareto front, while the other algorithms approaches the region around $f_1 = 0$ only slowly.

5.5 Comparison of crowding distance with the new archiving mechanism

A set of experiments was run to compare μARMOGA with crowding distance and μARMOGA with the new archiving mechanism. The population of four individuals was selected for the comparison. Results for problems DTLZ1, DTLZ2, DTLZ4, and WFG1 are summarized in Tables 22–29 and Figures 17–28.

According to these experiments, the new archiving approach outperforms crowding distance in diversity as is clear from Figs. 17–28 and spacing metric in Tabs. 22–29. While it also has very positive effect on the convergence of μARMOGA to the exact Pareto front for problems DTLZ1, DTLZ2, and DTLZ4 (Tabs. 22–27), both algorithms exhibit similar convergence for problem WFG1 (Tabs. 28 and 29).

5.6 Summary

As can be seen from above, μARMOGA outperforms the other methods in distribution of individuals along the Pareto front, and in many cases achieves

the best convergence as well. However, it should be noted that the default settings of the algorithms from PISA package is used, which may not be optimal for the test problems considered.

While IBEA offers exceptional convergence in some cases, the distribution of individuals along the exact Pareto front is usually rather poor, with many individuals attached to limits of the objective space. Our study confirms that the mechanism of crowding distance does not lead to uniform distribution of individuals along the Pareto front for more than two objectives. The same result might be observed from the comparison of μ ARMOGA using the two archiving mechanisms – crowding distance and the new proposed technique (Tables 22–29, and Figures 17–28). On the other hand, SPEA2 produces very uniform distribution of individuals comparable with μ ARMOGA in some instances.

Concerning the number of evaluations, DTLZ2 is the only problem, for which only 4 000 evaluations are sufficient to achieve reasonable convergence and distribution of individuals on the Pareto front by all algorithms. On the other hand, for DTLZ1, even 40 000 evaluations do not suffice to reach the true Pareto front for all seeds by any approach, and results for 100 000 and 200 000 evaluations are added for a reasonable comparison. Even this large number of evaluations was not sufficient to reach the proximity of exact front in the case of problem WFG1, and results for 1 000 000 and 2 000 000 evaluations are added. For test functions DTLZ4 and WFG1, the new archiving mechanism is able to drive the evolution to regions, where the coverage of the Pareto front by individuals is sparse, and recover nice distribution of individuals along the Pareto set even for poorly chosen initial population.

To investigate the optimal distribution of the number of function evaluations between population size and number of generations for micro-evolution, μ ARMOGA is run with 4, 10 and 20 individuals for DTLZ1, DTLZ2, and

DTLZ4. According to our experiments, the performance of the algorithm is similar for all configurations with respect to spacing and convergence history and no strong dependence is revealed. However, for problem DTLZ4, the method tends to produce more degenerated fronts with larger population (see Table 21). Additionally, population of 4 individuals leads to the best convergence metrics for problem DTLZ1, and population of 20 individuals to the best converged front for problem DTLZ4. Thus, using small populations and larger number of generations seems as the preferable approach.

6 Conclusion

The goal of our study is twofold: (a) to develop a new approach for selecting individuals to the Pareto archive; (b) to explore the potential of using small population in evolutionary algorithms.

The main contribution of the paper is the presentation of a new archiving mechanism. Although its basic idea is rather simple and straightforward, the technique produces very promising results on all tested problems. We are aware of the fact that the theoretical time complexity of the mechanism might be rather large (quadratic in the worst case). However, our tests justify its usage, since the experimentally found complexity is much more favourable (approximately linear). Moreover, it is intended to be used in combination with small population, for which such more elaborate selection mechanism is usually affordable.

The proposed selection mechanism was combined with μ ARMOGA and is compared to other three state-of-the-art algorithms (NSGA-II, SPEA2, and IBEA) on four test problems. We can conclude that μ ARMOGA presents Pareto sets with the same or better distribution as SPEA2, but usually with much better convergence to the exact front that is comparable with IBEA, thus the best

combining requirements on both convergence and distribution of individuals. A considerable improvement is attained, using the new mechanism, in comparison with the version of μ ARMOGA that uses crowding distance. Clearly, μ ARMOGA equipped with the new diversity mechanism is very promising and may be competitive with respect to other recent approaches.

Our experiments further support using small populations (up to 10 individuals), since runs with four individuals usually produces the best results. It is well known that such small population can lead to rapid convergence. However, in combination with the proposed archiving mechanism, it also seems to be more robust with respect to an initial population.

Regarding the history of convergence to the Pareto front, in some cases as few as 4 000 evaluations of objective function could be sufficient for some problems (DTLZ2), while for other problems (multi-modal problem DTLZ1 or difficult WFG1), even 40 000 evaluations may not be sufficient to approximate the true Pareto front, and as many as 1 000 000 evaluations are needed for reasonable outcome.

Acknowledgement

This research has been supported by the Development of Applied External Aerodynamics Program (Ministry of Education, Youth and Sports of the Czech Republic Grant MSM0001066901), by research project AV0Z10190503 (Academy of Sciences of the Czech Republic), and by grant IAA100760702 (Grant Agency of the Academy of Sciences of the Czech Republic).

References

- [1] ARAKAWA, M., AND HAGIWARA, I. Development of adaptive real range

- (ARRange) genetic algorithms. *JSME International Journal Series C, Mechanical systems, machine elements and manufacturing* 41, 4 (1998), 969–977.
- [2] BLEULER, S., LAUMANN, M., THIELE, L., AND ZITZLER, E. PISA – A platform and programming language independent interface for search algorithms. In *Evolutionary Multi-Criterion Optimization (EMO 2003)* (Berlin, 2003), C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds., Lecture Notes in Computer Science, Springer, pp. 494–508.
- [3] COELLO, C. A. C., AND PULIDO, G. T. A micro-genetic algorithm for multiobjective optimization. In *Evolutionary Multi-Criterion Optimization (EMO 2001)* (Berlin, 2001), E. Zitzler, K. Deb, L. Thiele, C. Coello Coello, and D. Corne, Eds., Lecture Notes in Computer Science, Springer, pp. 126–140.
- [4] COELLO, C. A. C., PULIDO, G. T., AND LECHUGA, M. S. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8, 3 (2004), 256–279.
- [5] DEB, K. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.
- [6] DEB, K., AGRAWAL, S., PRATAP, A., AND MEYARIVAN, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature – PPSN VI* (Berlin, 2000), M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds., Springer, pp. 849–858.

- [7] DEB, K., MOHAN, M., AND MISHRA, S. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. Tech. Rep. 2003002, KanGAL, Indian Institute of Technology Kanpur, India, 2003.
- [8] DEB, K., THIELE, L., LAUMANN, M., AND ZITZLER, E. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC '02)* (Piscataway, New Jersey, 2002), IEEE Press, pp. 825–830.
- [9] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.
- [10] HUBAND, S., BARONE, L., WHILE, R. L., AND HINGSTON, P. A scalable multi-objective test problem toolkit. In *EMO (2005)*, C. A. C. Coello, A. H. Aguirre, and E. Zitzler, Eds., vol. 3410 of *Lecture Notes in Computer Science*, Springer, pp. 280–295.
- [11] KNOWLES, J. D., AND CORNE, D. W. The Pareto archived evolution strategy : A new baseline algorithm for Pareto multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)* (1999), vol. 1, pp. 98–105.
- [12] KNOWLES, J. D., AND CORNE, D. W. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation* 8, 2 (2000), 149–172.
- [13] KRISHNAKUMAR, K. Micro-genetic algorithms for stationary and non-stationary function optimization. In *SPIE's Intelligent Control and Adaptive Systems Conference* (1989), G. Rodriguez, Ed., Society of Photo-Optical Instrumentation Engineers (SPIE), pp. 289–296.
- [14] OYAMA, A. *Wing Design Using Evolutionary Algorithm*. PhD thesis, Tohoku University, Japan, 2000.

- [15] SZÖLLÖS, A., ŠMÍD, M., AND HÁJEK, J. Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and ϵ -dominance. *Advances in Engineering Software* 40, 6 (2009), 419–430.
- [16] VAN VELDHUIZEN, D. A., AND LAMONT, G. B. Multiobjective evolutionary algorithm research: A history and analysis. Tech. Rep. TR-98-03, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2001.
- [17] ZITZLER, E., AND KÜNZLI, S. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature (PPSN VIII)* (Berlin, 2004), X. Yao et al., Eds., Lecture Notes in Computer Science, Springer, pp. 832–842.
- [18] ZITZLER, E., LAUMANN, M., AND THIELE, L. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems. Proceedings of the EUROGEN2001 Conference* (Barcelona, 2002), K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papaliliou, and T. Fogarty, Eds., International Center for Numerical Methods in Engineering (CIMNE), pp. 95–100.
- [19] ZITZLER, E., AND THIELE, L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.

metric	NSGA-II	SPEA2	IBEA
GD	3.61e+01	2.99e+01	1.80e+00
TOL5	7.53e+01	5.92e+01	2.10e+00
spacing	2.47e+00	2.82e+00	2.14e+00
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	4.17e+00	1.13e+01	1.08e+01
TOL5	5.70e+00	1.50e+01	1.50e+01
spacing	7.39e-01	7.40e-01	8.45e-01

Table 1: Problem DTLZ1, 4 000 function evaluations

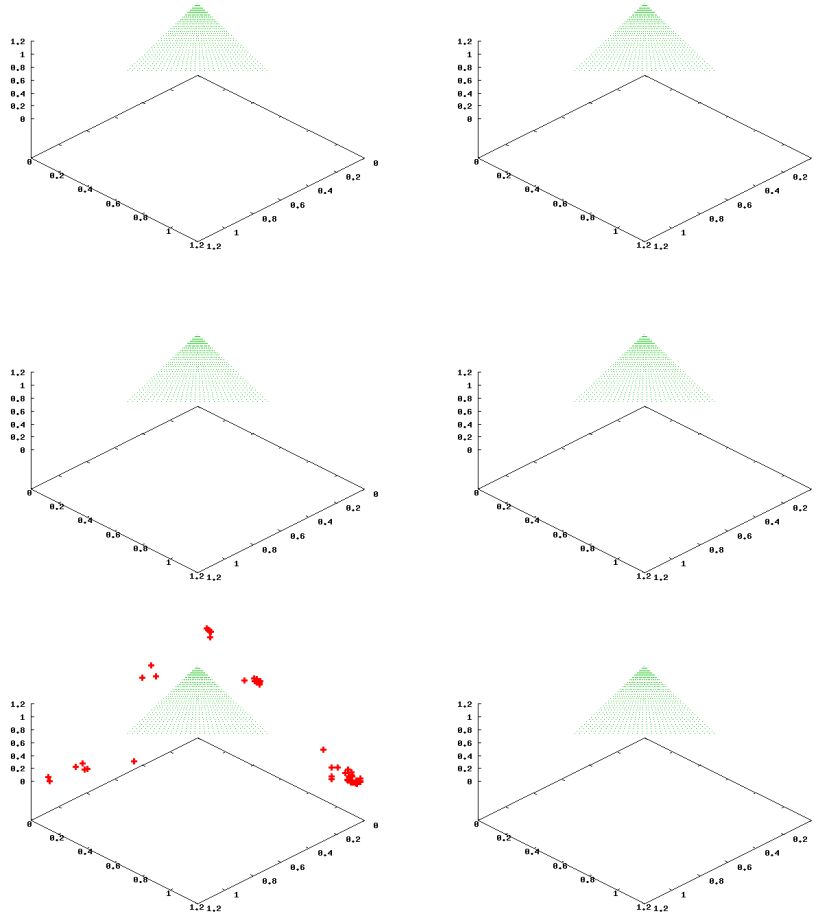


Figure 1: Pareto front after 4 000 function evaluations, problem DTLZ1, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	4.96e+00	3.32e+00	5.91e-01
TOL5	8.52e+00	5.80e+00	6.64e-01
spacing	1.50e+00	2.32e+00	2.65e+00
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	2.35e-01	6.10e+00	4.21e+00
TOL5	3.05e-01	7.44e+00	5.33e+00
spacing	2.63e-01	9.85e-01	8.24e-01

Table 2: Problem DTLZ1, 20 000 function evaluations

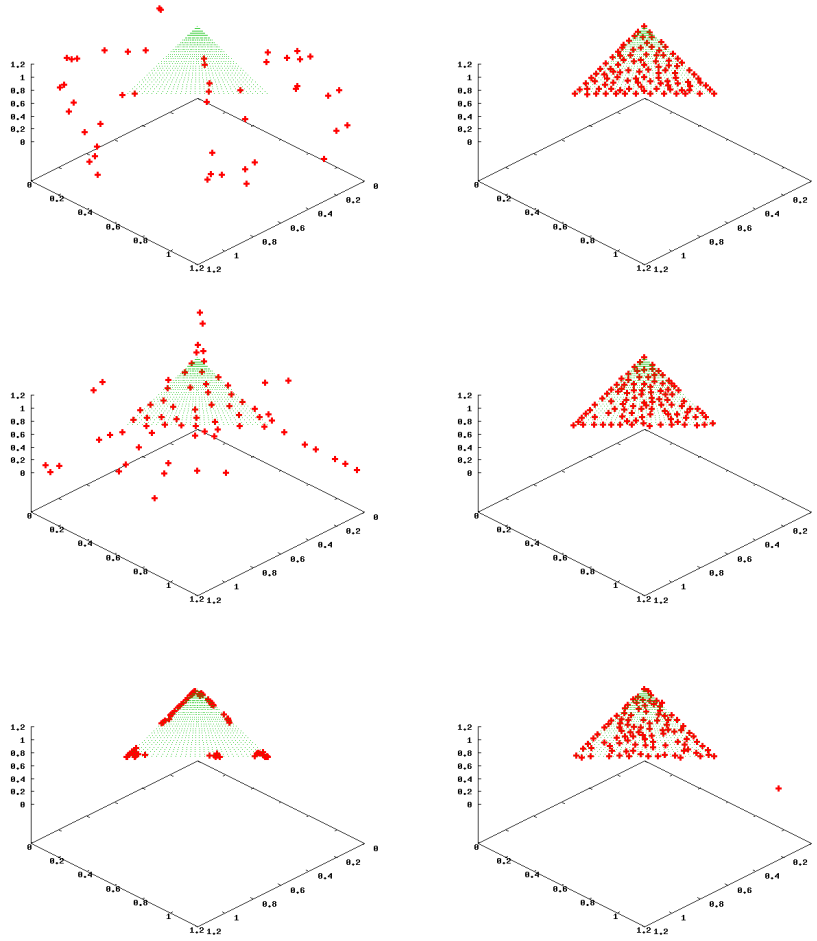


Figure 2: Pareto front after 20 000 function evaluations, problem DTLZ1, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	3.82e+00	2.31e+00	4.93e-01
TOL5	6.36e+00	3.71e+00	4.46e-01
spacing	1.56e+00	1.71e+00	4.19e+00
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	3.61e-02	4.77e+00	3.02e+00
TOL5	5.04e-02	6.00e+00	3.79e+00
spacing	1.39e-01	4.15e-01	1.70e-01

Table 3: Problem DTLZ1, 40 000 function evaluations

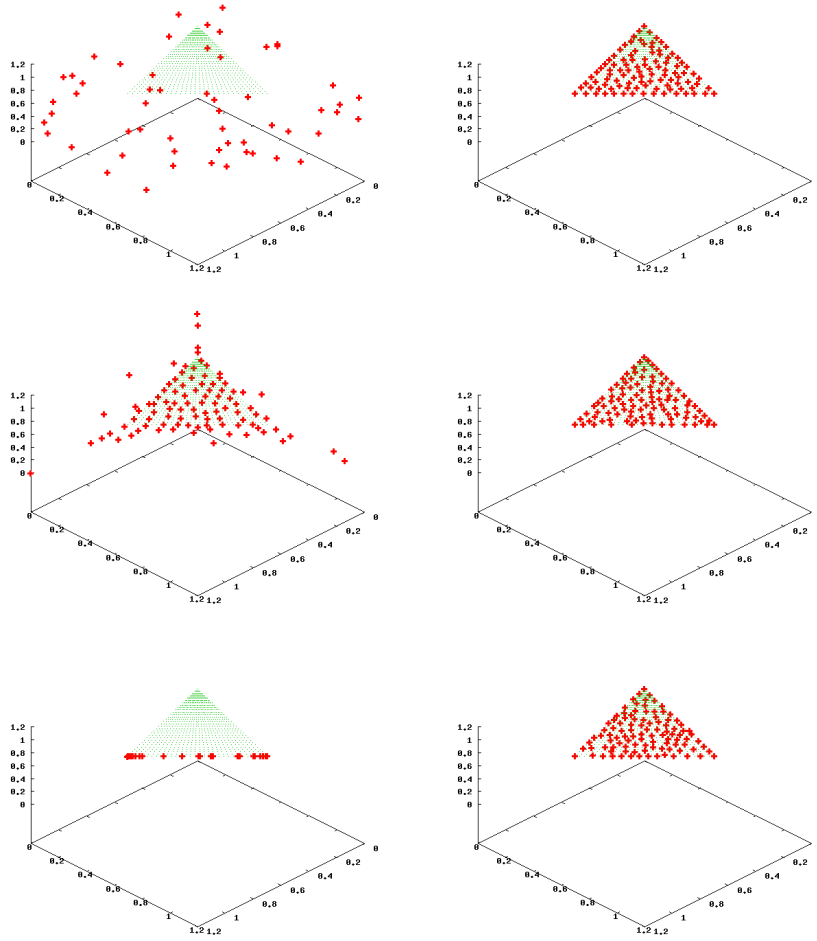


Figure 3: Pareto front after 40 000 function evaluations, problem DTLZ1, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	2.33e+00	1.03e+00	-
TOL5	4.01e+00	2.01e+00	-
spacing	1.51e+00	1.42e+00	-
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	1.88e-03	4.60e+00	1.91e+00
TOL5	4.48e-04	5.82e+00	2.41e+00
spacing	8.50e-02	9.10e-02	1.51e-01

Table 4: Problem DTLZ1, 100 000 function evaluations

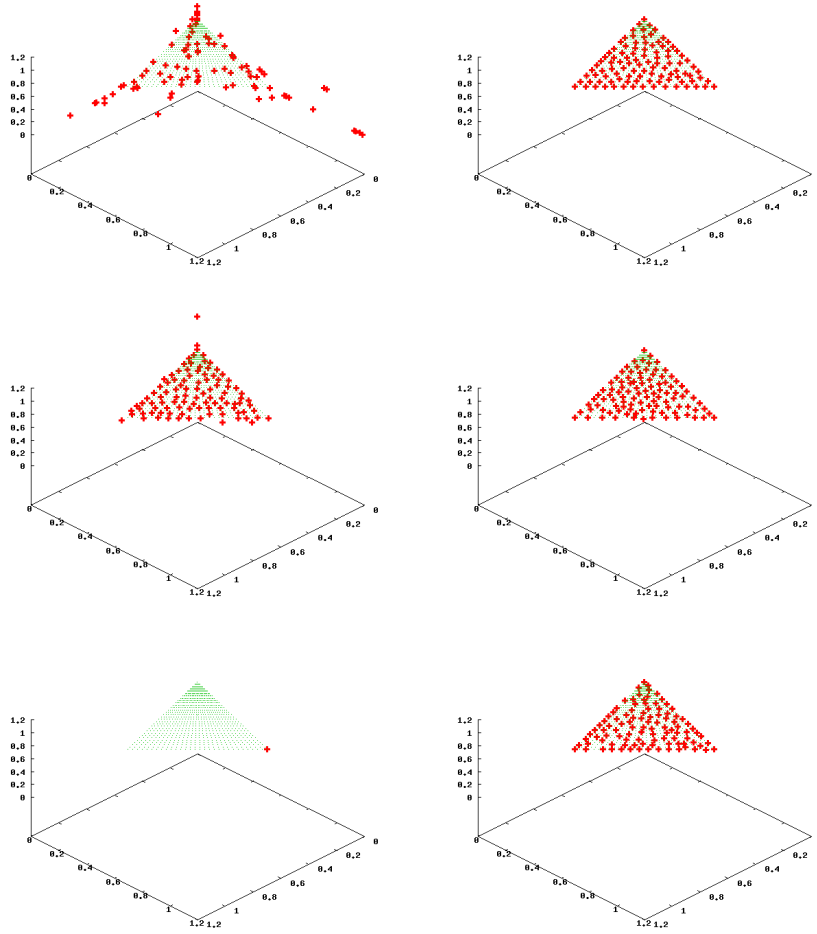


Figure 4: Pareto front after 100 000 function evaluations, problem DTLZ1, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	2.26e+00	3.76e-01	-
TOL5	2.35e+00	5.64e-01	-
spacing	1.43e+00	1.17e+00	-
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	1.54e-03	4.59e+00	2.08e+00
TOL5	1.74e-04	5.82e+00	2.30e+00
spacing	7.82e-02	8.33e-02	5.19e-01

Table 5: Problem DTLZ1, 200 000 function evaluations

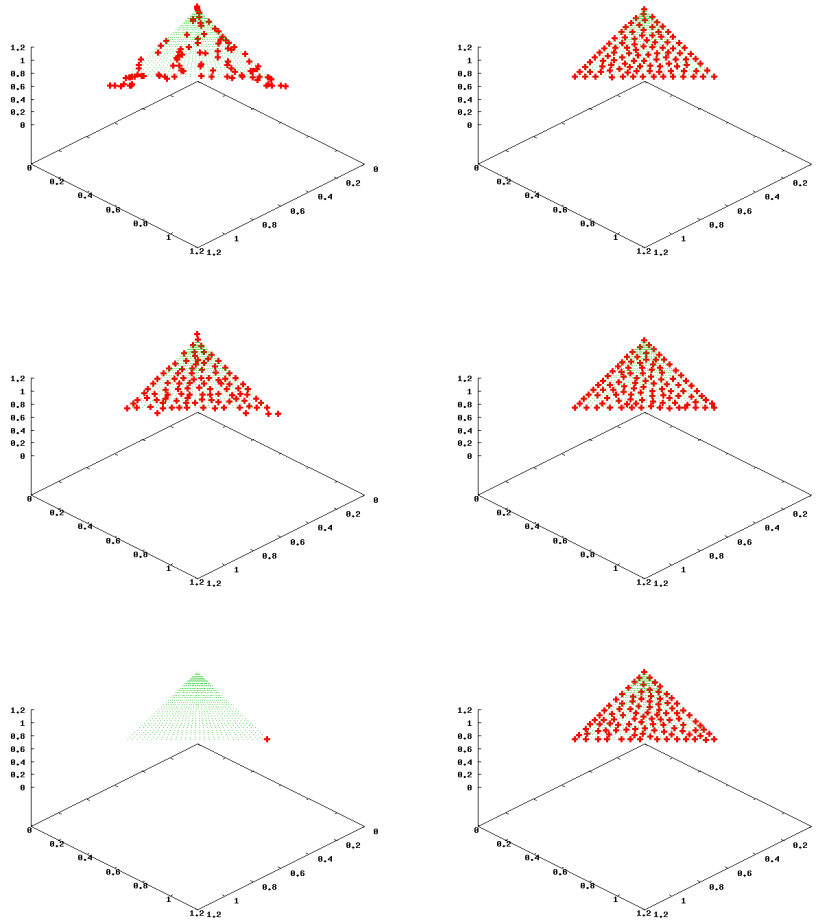


Figure 5: Pareto front after 200 000 function evaluations, problem DTLZ1, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	1.07e-01	9.08e-02	4.96e-03
TOL5	2.51e-01	2.05e-01	7.49e-03
spacing	6.40e-01	1.83e-01	7.44e-01
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	1.41e-02	1.77e-02	1.89e-02
TOL5	2.82e-02	3.77e-02	3.62e-02
spacing	1.30e-01	1.64e-01	1.80e-01

Table 6: Problem DTLZ2, 4000 function evaluations

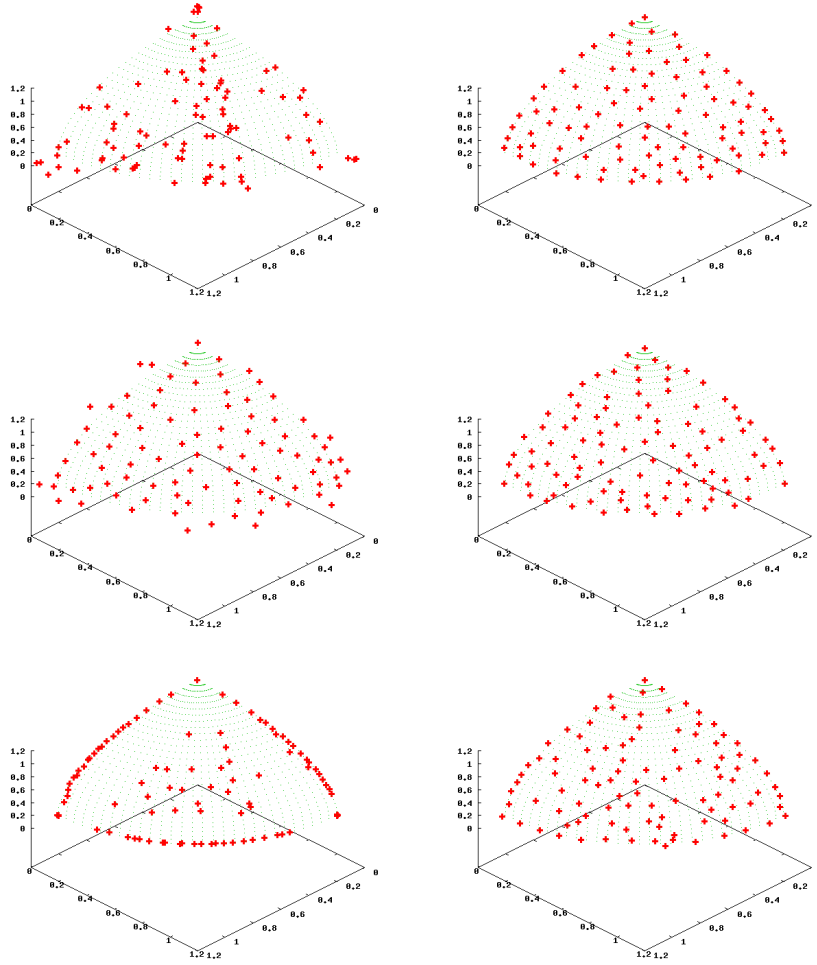


Figure 6: Pareto front after 4000 function evaluations, problem DTLZ2, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	6.29e-02	3.50e-02	4.15e-04
TOL5	1.49e-01	7.53e-02	4.97e-04
spacing	6.45e-01	1.40e-01	6.80e-01
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	2.59e-03	1.99e-03	2.94e-03
TOL5	4.01e-03	3.72e-03	3.29e-03
spacing	6.94e-02	8.09e-02	8.75e-02

Table 7: Problem DTLZ2, 20 000 function evaluations

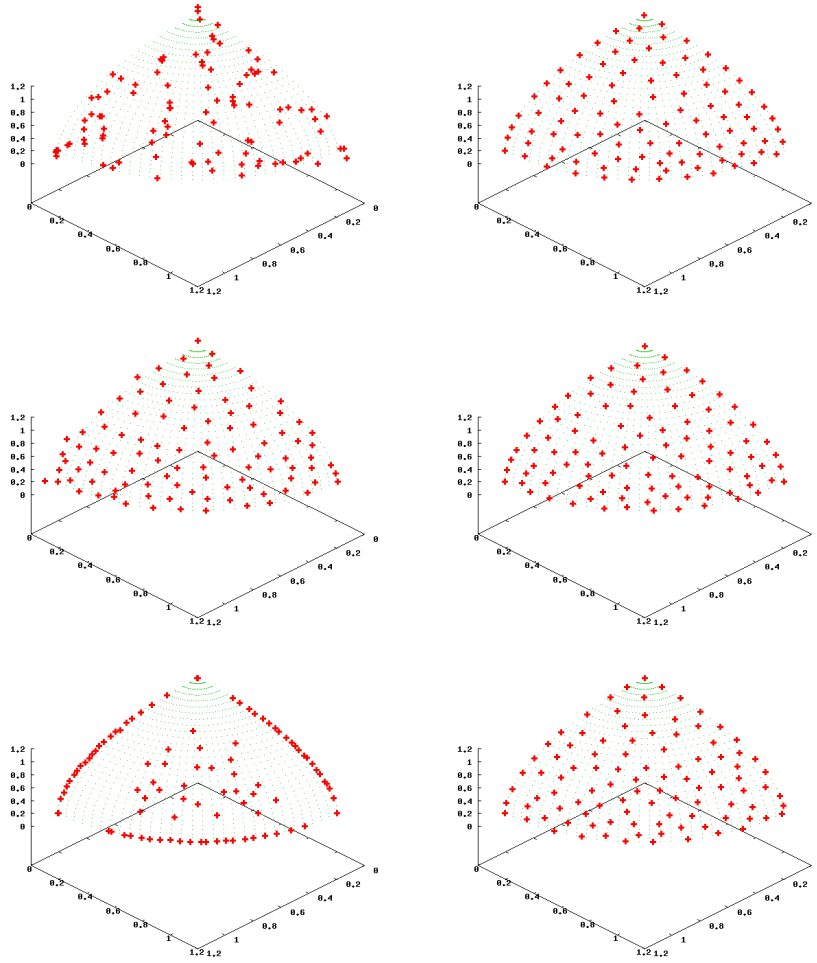


Figure 7: Pareto front after 20 000 function evaluations, problem DTLZ2, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	4.64e-02	1.63e-02	4.94e-04
TOL5	1.06e-01	3.50e-02	2.71e-04
spacing	6.14e-01	1.33e-01	6.82e-01
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	1.04e-03	1.32e-03	1.02e-03
TOL5	9.23e-04	1.02e-03	1.06e-03
spacing	6.03e-02	6.71e-02	7.03e-02

Table 8: Problem DTLZ2, 40 000 function evaluations

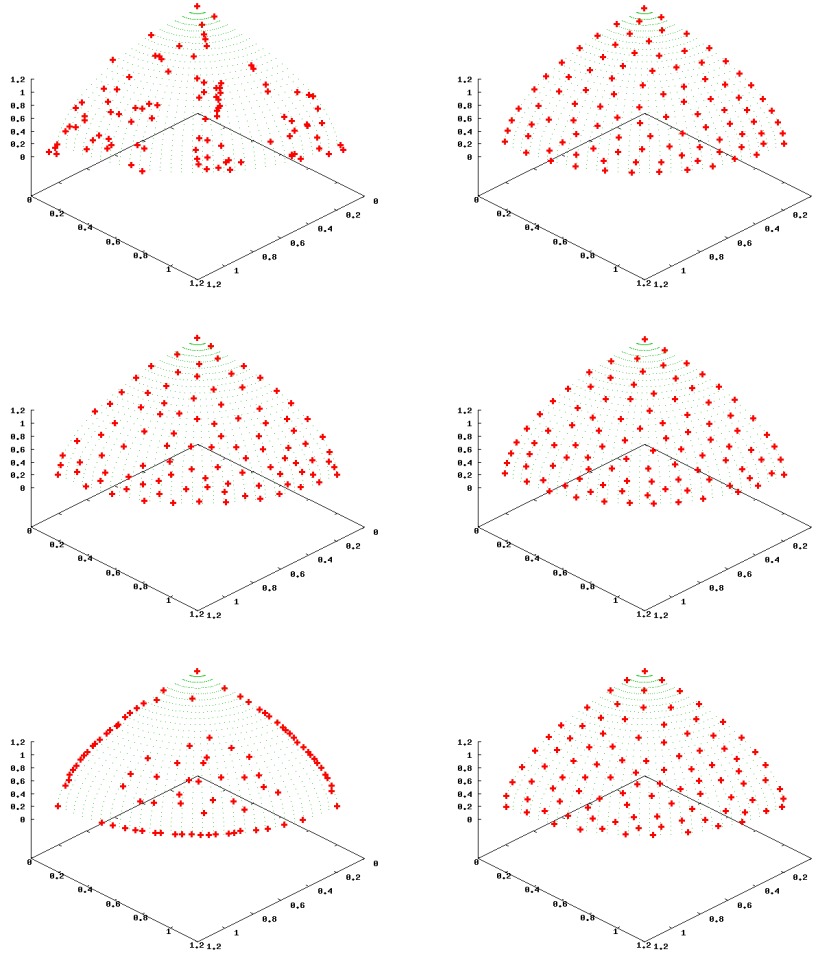


Figure 8: Pareto front after 40 000 function evaluations, problem DTLZ2, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	8.91e-02	1.31e-01	4.78e-03
TOL5	2.01e-01	2.81e-01	7.32e-03
spacing	6.13e-01	1.80e-01	7.30e-01
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	1.87e-03	1.27e-02	5.45e-03
TOL5	3.42e-03	2.51e-02	1.23e-02
spacing	8.48e-01	1.88e+00	2.09e+00

Table 9: Problem DTLZ4, 4000 function evaluations

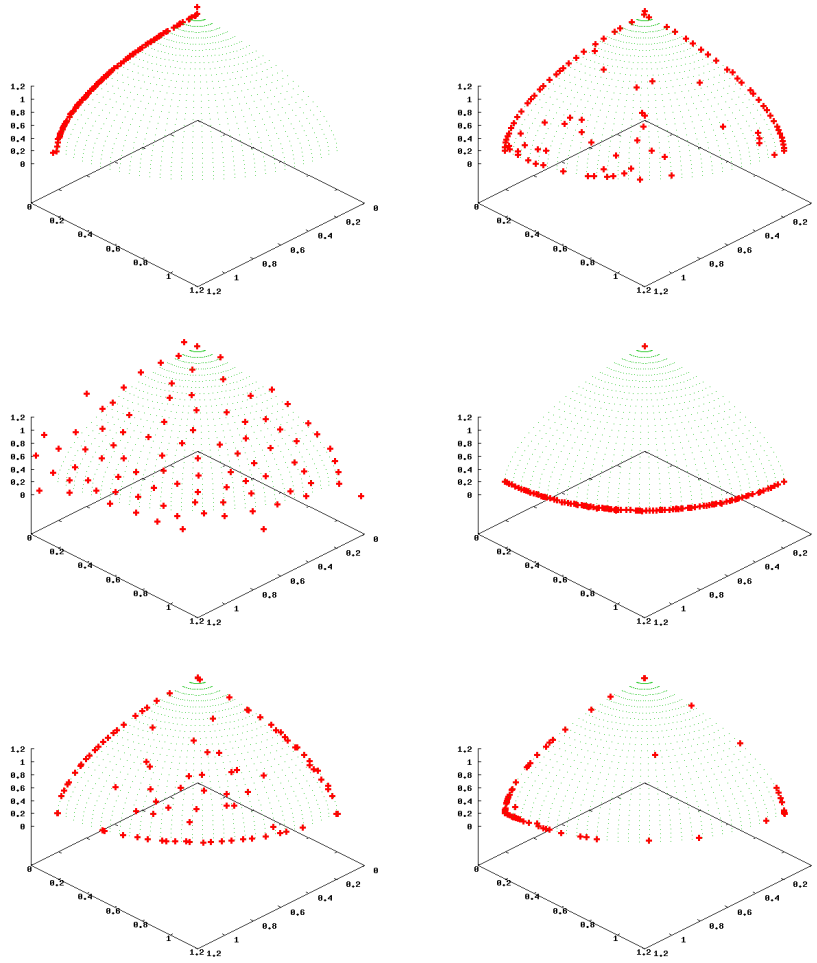


Figure 9: Pareto front after 4000 function evaluations, problem DTLZ4, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	2.61e-02	3.92e-02	5.60e-04
TOL5	6.25e-02	9.73e-02	2.87e-04
spacing	6.18e-01	1.21e-01	6.83e-01
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	4.34e-04	2.34e-04	3.55e-05
TOL5	1.43e-04	1.21e-04	3.40e-05
spacing	1.46e-01	1.12e-01	3.45e-01

Table 10: Problem DTLZ4, 20 000 function evaluations

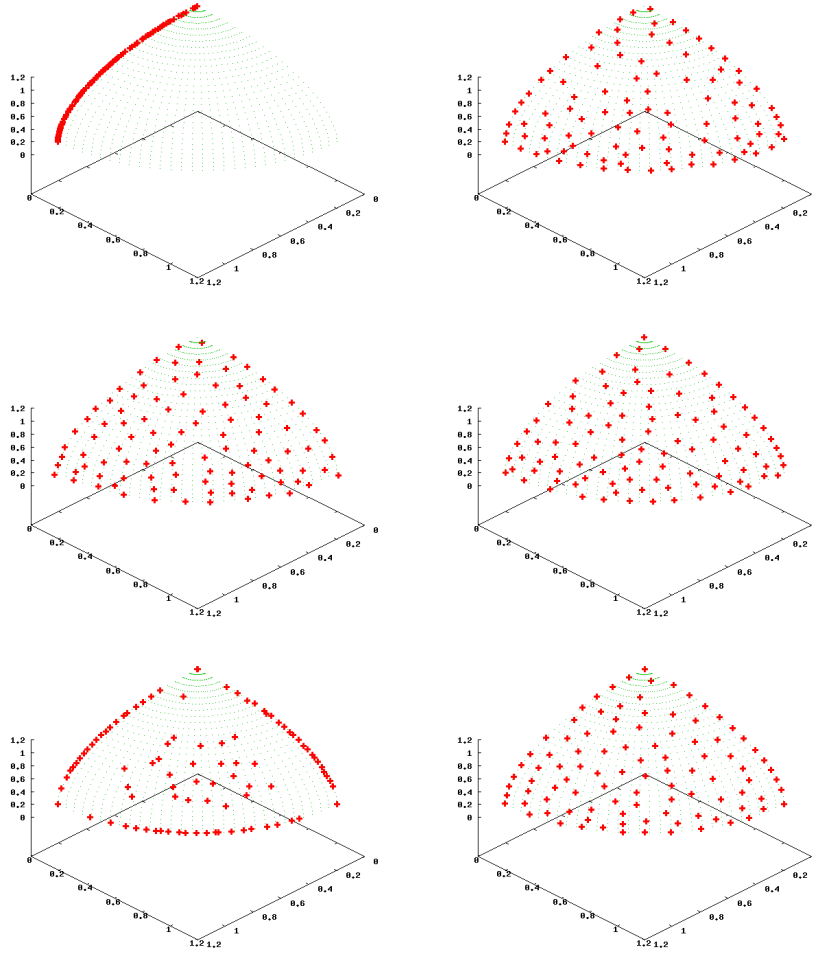


Figure 10: Pareto front after 20 000 function evaluations, problem DTLZ4, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA
GD	1.61e-02	1.91e-02	1.75e-04
TOL5	2.94e-02	3.60e-02	1.39e-04
spacing	6.42e-01	1.33e-01	6.94e-01
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
GD	2.19e-04	3.81e-04	1.44e-05
TOL5	3.48e-05	2.78e-05	1.15e-05
spacing	9.36e-02	8.01e-02	9.55e-02

Table 11: Problem DTLZ4, 40 000 function evaluations

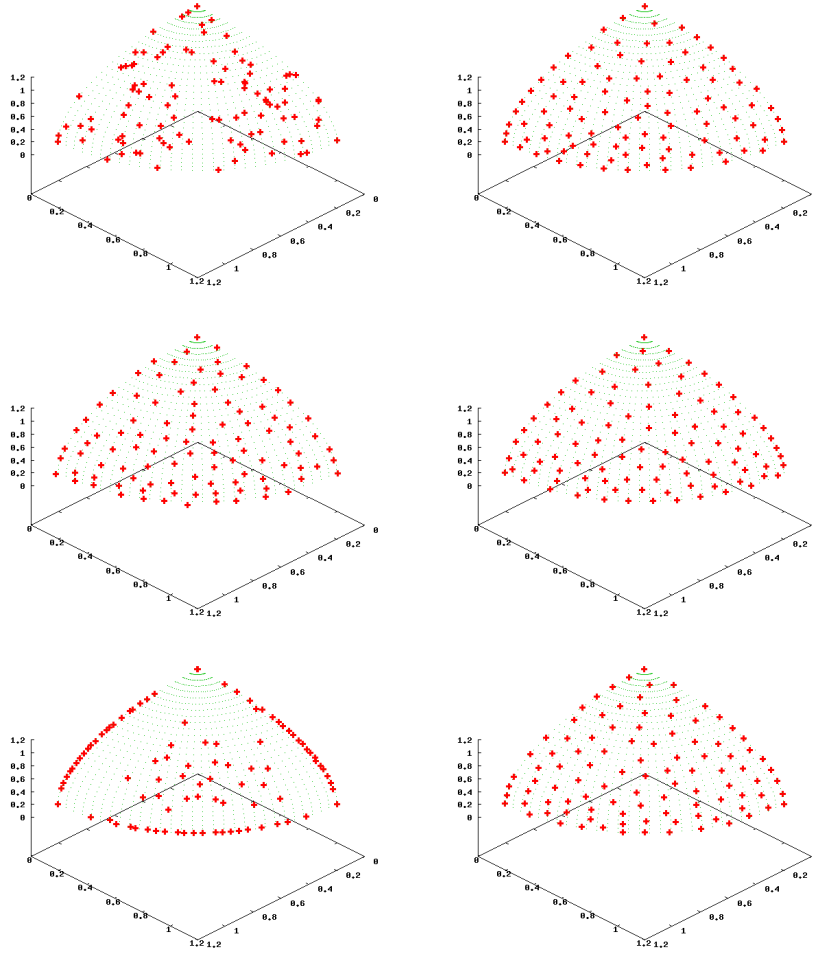


Figure 11: Pareto front after 40 000 function evaluations, problem DTLZ4, NSGA-II (top left), SPEA2 (centre left), IBEA (bottom left), and μ ARMOGA with population size 4 (top right), 10 (centre right), 20 (bottom right).

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	8.86e-01	8.82e-01	7.47e-01	7.95e-01
TOL5	1.05e+00	1.04e+00	8.40e-01	9.27e-01
spacing	4.90e-01	1.60e-01	5.40e-01	5.19e-01

Table 12: Problem WFG1, 4 000 function evaluations

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	9.01e-01	8.95e-01	7.67e-01	4.97e-01
TOL5	1.06e+00	1.05e+00	8.62e-01	5.92e-01
spacing	4.88e-01	1.67e-01	5.66e-01	2.82e-01

Table 13: Problem WFG1, 20 000 function evaluations

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	8.86e-01	8.82e-01	7.47e-01	3.77e-01
TOL5	1.05e+00	1.04e+00	8.40e-01	4.56e-01
spacing	4.90e-01	1.60e-01	5.40e-01	2.24e-01

Table 14: Problem WFG1, 40 000 function evaluations

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	8.80e-01	8.75e-01	7.26e-01	2.51e-01
TOL5	1.05e+00	1.04e+00	8.15e-01	3.15e-01
spacing	4.94e-01	1.62e-01	5.29e-01	1.96e-01

Table 15: Problem WFG1, 100 000 function evaluations

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	8.75e-01	8.79e-01	7.10e-01	1.75e-01
TOL5	1.04e+00	1.04e+00	7.99e-01	2.29e-01
spacing	4.98e-01	1.66e-01	5.47e-01	1.86e-01

Table 16: Problem WFG1, 200 000 function evaluations

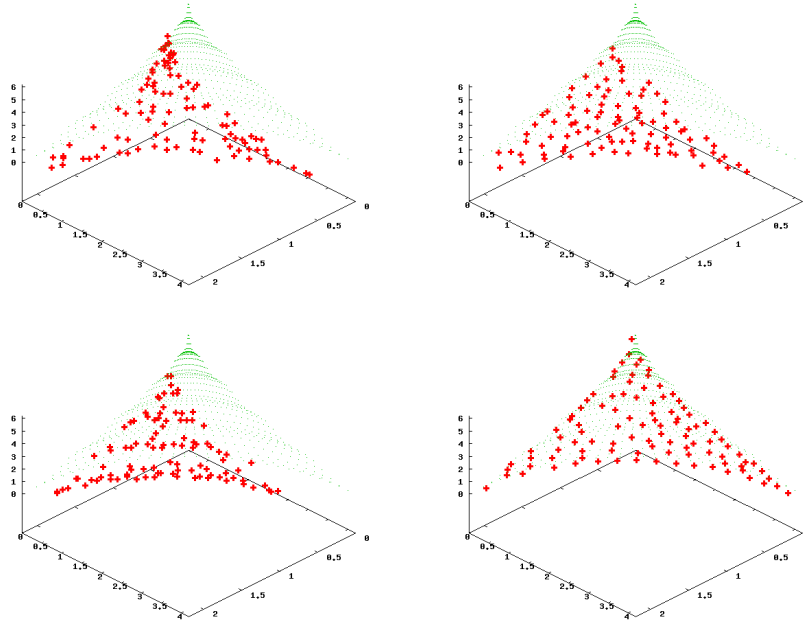


Figure 12: Pareto front after 200 000 function evaluations, problem WFG1, NSGA-II (top left), SPEA2 (top right), IBEA (bottom left), and μ ARMOGA with population size 4 (bottom right).

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	8.58e-01	8.71e-01	6.81e-01	6.53e-02
TOL5	1.03e+00	1.03e+00	7.66e-01	1.14e-01
spacing	4.72e-01	1.74e-01	5.24e-01	1.53e-01

Table 17: Problem WFG1, 1 000 000 function evaluations

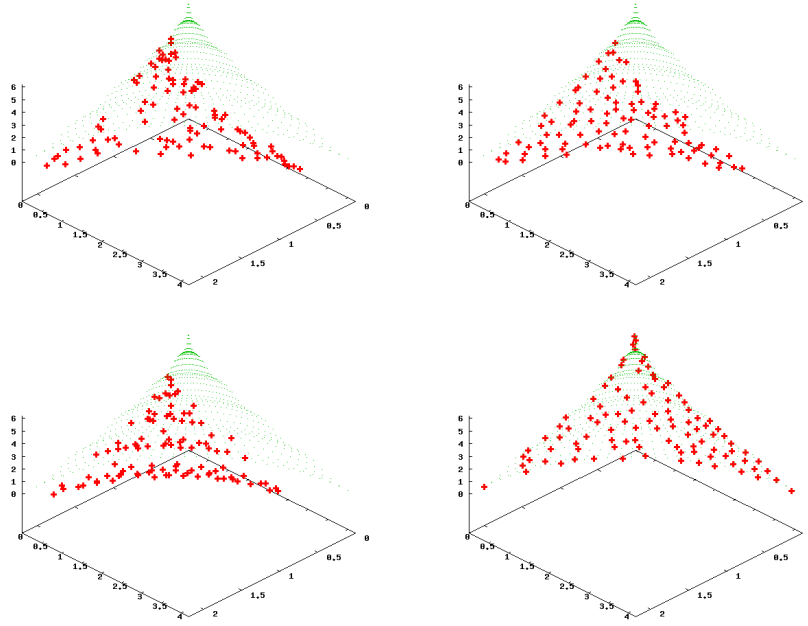


Figure 13: Pareto front after 1 000 000 function evaluations, problem WFG1, NSGA-II (top left), SPEA2 (top right), IBEA (bottom left), and μ ARMOGA with population size 4 (bottom right).

metric	NSGA-II	SPEA2	IBEA	μ ARMOGA(4)
GD	8.58e-01	8.74e-01	6.69e-01	4.79e-02
TOL5	1.03e+00	1.03e+00	7.52e-01	9.69e-02
spacing	5.06e-01	1.89e-01	5.35e-01	1.54e-01

Table 18: Problem WFG1, 2 000 000 function evaluations

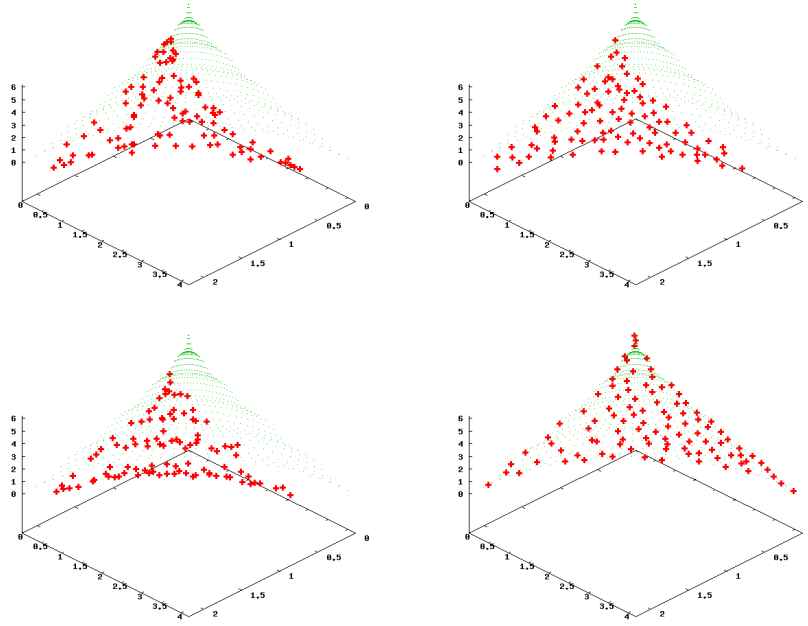


Figure 14: Pareto front after 2 000 000 function evaluations, problem WFG1, NSGA-II (top left), SPEA2 (top right), IBEA (bottom left), and μ ARMOGA with population size 4 (bottom right).

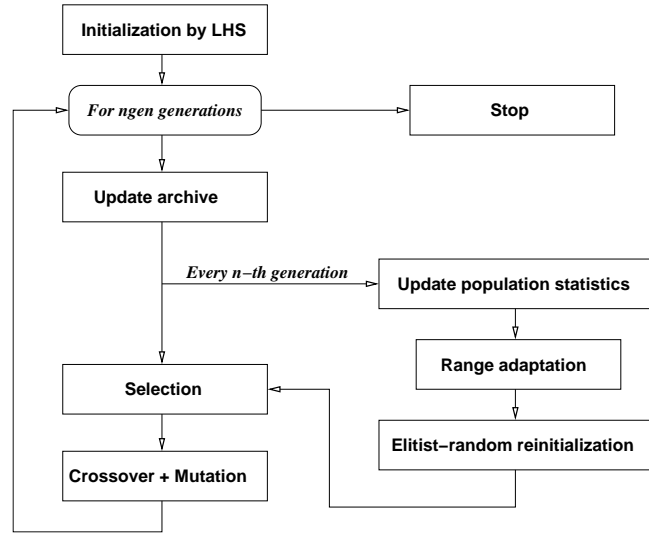


Figure 15: Simple scheme of the μ ARMOGA algorithm.

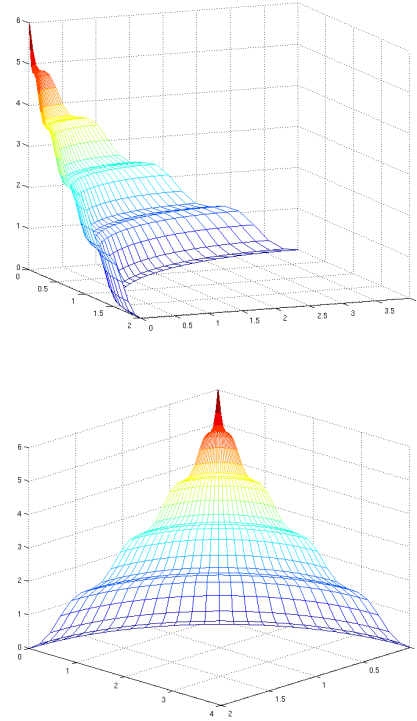


Figure 16: Exact Pareto front for problem WFG1.

metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
DTLZ1	0.98	1.32	1.32
DTLZ2	0.99	1.42	1.47
DTLZ4	1.05	1.35	1.35
WFG1	1.08	n/a	n/a

Table 19: Average number of updates necessary after the addition of an individual into archive, archive size limit 100.

archive size	20	50	100	200	500	1 000
DTLZ4	1.06	1.05	1.06	1.06	1.02	0.94

Table 20: Average number of updates necessary after the addition of an individual into archive for problem DTLZ4, variable archive size.

metric	NSGA-II	SPEA2	IBEA
DTLZ1	-/-/-/-/-	-/-/-/-/-	-/2/11/20/20
DTLZ2	-/-/-	-/-/-	-/-/-
DTLZ4	11/8/5	11/11/11	6/6/6
WFG1	-/-/-/-/-/-/-	-/-/-/-/-/-/-	-/-/-/-/-/-/-
metric	μ ARMOGA(4)	μ ARMOGA(10)	μ ARMOGA(20)
DTLZ1	-/-/-/-/-	-/-/-/-/-	-/-/-/-/-
DTLZ2	-/-/-	-/-/-	-/-/-
DTLZ4	-/-/-	4/4/4	4/3/3
WFG1	-/-/-/-/-/-/-	n/a	n/a

Table 21: Number of degenerated Pareto fronts for 20 seeds. Number at 4 000/20 000/40 000(/100 000/200 000(/1 000 000/2 000 000)) function evaluations.

evaluations	4 000	20 000	40 000	100 000	200 000
GD	2.13e+01	2.28e+00	1.15e+00	2.41e+00	3.72e-01
TOL5	4.02e+01	1.42e+00	2.07e-01	3.99e-02	1.22e-02
spacing	3.34e+00	2.34e+00	2.73e+00	3.29e+00	2.60e+00
degenerated	0	0	0	0	0

Table 22: μ ARMOGA with crowding distance, four individuals, problem DTLZ1, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000
GD	4.17e+00	2.35e-01	3.61e-02	1.88e-03	1.54e-03
TOL5	5.70e+00	3.05e-01	5.04e-02	4.48e-04	1.74e-04
spacing	7.39e-01	2.63e-01	1.39e-01	8.50e-02	7.82e-02
degenerated	0	0	0	0	0

Table 23: μ ARMOGA with the new proposed archiving algorithm, four individuals, problem DTLZ1, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000
GD	5.36e-02	2.54e-03	1.77e-03	2.16e-03	1.99e-03
TOL5	7.92e-02	2.92e-03	2.83e-03	3.58e-03	2.83e-01
spacing	5.74e-01	5.60e-01	5.47e-01	5.49e-01	5.41e-01
degenerated	0	0	0	0	0

Table 24: μ ARMOGA with crowding distance, four individuals, problem DTLZ2, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000
GD	1.41e-02	2.59e-03	1.04e-03	3.45e-04	1.81e-04
TOL5	2.82e-02	4.01e-03	9.23e-04	1.09e-04	2.39e-05
spacing	1.30e-01	6.94e-02	6.03e-02	4.94e-02	4.46e-02
degenerated	0	0	0	0	0

Table 25: μ ARMOGA with the new proposed archiving algorithm, four individuals, problem DTLZ2, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000
GD	5.46e+01	2.35e+01	1.18e+01	1.68e+01	1.48e+01
TOL5	1.46e+02	1.96e+01	1.92e-01	1.70e-01	1.67e-01
spacing	3.40e+00	6.50e+00	5.09e+00	4.73e+00	4.86e+00
degenerated	2	3	3	3	3

Table 26: μ ARMOGA with crowding distance, four individuals, problem DTLZ4, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000
GD	1.87e-03	4.34e-04	2.19e-04	9.32e-05	3.39e-05
TOL5	3.42e-03	1.43e-04	3.48e-05	1.34e-05	8.82e-06
spacing	8.48e-01	1.46e-01	9.36e-02	7.00e-02	6.35e-02
degenerated	0	0	0	0	0

Table 27: μ ARMOGA with the new proposed archiving algorithm, four individuals, problem DTLZ4, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000	1 000 000	2 000 000
GD	9.01e-01	3.67e-01	2.81e-01	1.93e-01	1.27e-01	4.09e-02	3.93e-02
TOL5	1.21e+00	4.26e-01	3.32e-01	2.37e-01	1.66e-01	8.01e-02	8.51e-02
spacing	1.46e+00	7.03e-01	6.85e-01	6.17e-01	6.24e-01	5.83e-01	5.28e-01
degenerated	0	0	0	0	0	0	0

Table 28: μ ARMOGA with crowding distance, four individuals, problem WFG1, average for twenty seeds.

evaluations	4 000	20 000	40 000	100 000	200 000	1 000 000	2 000 000
GD	7.95e-01	4.97e-01	3.77e-01	2.51e-01	1.75e-01	6.53e-02	4.79e-02
TOL5	9.27e-01	5.92e-01	4.56e-01	3.15e-01	2.29e-01	1.14e-01	9.69e-02
spacing	5.19e-01	2.82e-01	2.24e-01	1.96e-01	1.86e-01	1.53e-01	1.54e-01
degenerated	0	0	0	0	0	0	0

Table 29: μ ARMOGA with the new proposed archiving algorithm, four individuals, problem WFG1, average for twenty seeds.

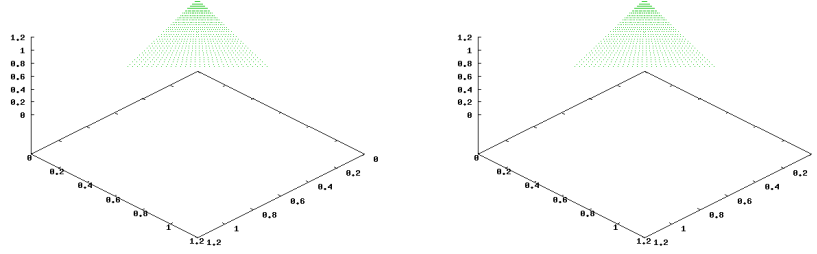


Figure 17: Pareto front after 4000 function evaluations, problem DTLZ1, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

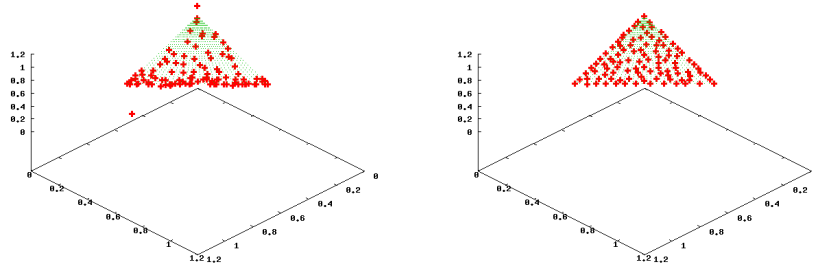


Figure 18: Pareto front after 20000 function evaluations, problem DTLZ1, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

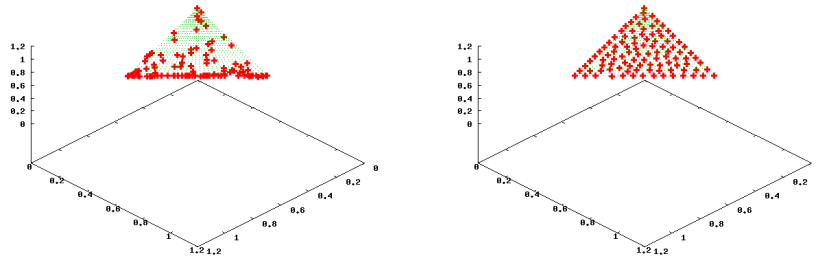


Figure 19: Pareto front after 200000 function evaluations, problem DTLZ1, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

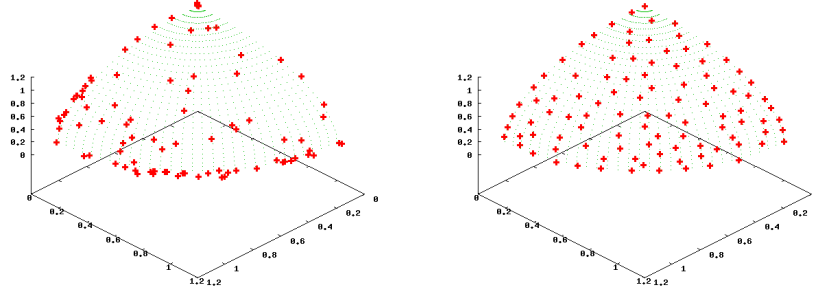


Figure 20: Pareto front after 4000 function evaluations, problem DTLZ2, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

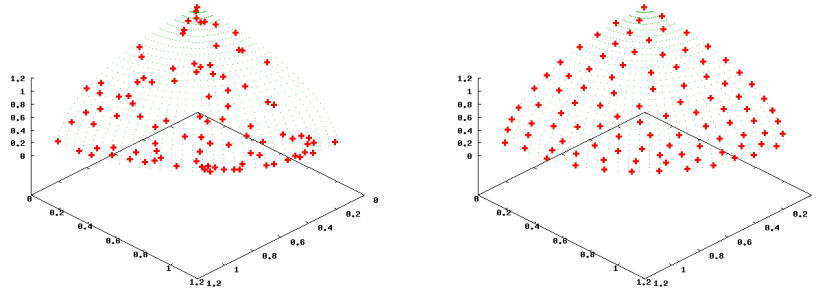


Figure 21: Pareto front after 20000 function evaluations, problem DTLZ2, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

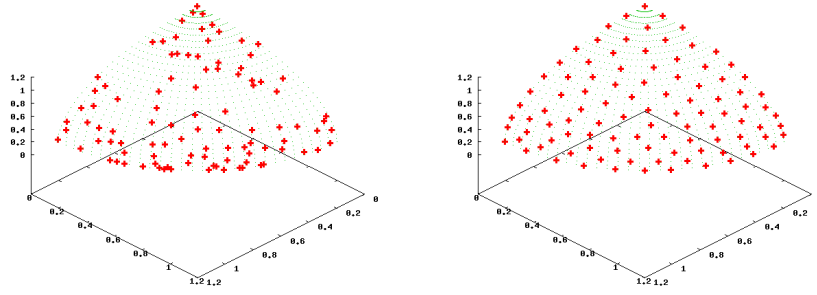


Figure 22: Pareto front after 200000 function evaluations, problem DTLZ2, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

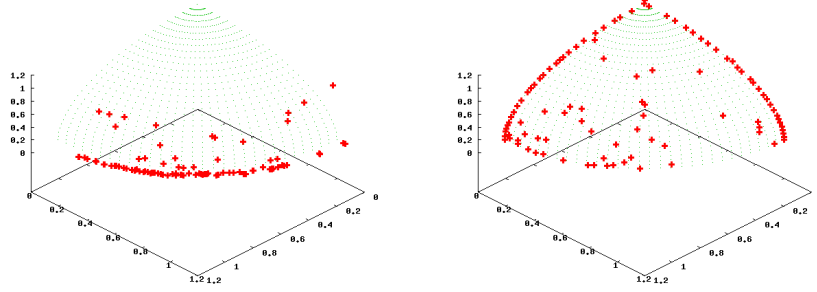


Figure 23: Pareto front after 4000 function evaluations, problem DTLZ4, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

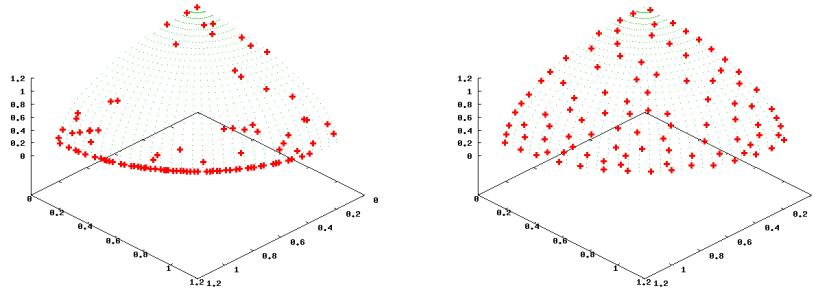


Figure 24: Pareto front after 20000 function evaluations, problem DTLZ4, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

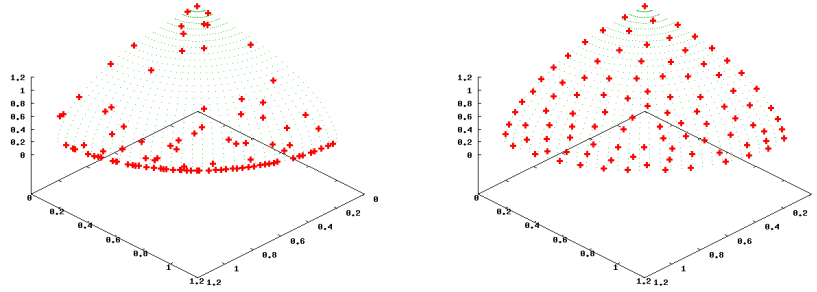


Figure 25: Pareto front after 200000 function evaluations, problem DTLZ4, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

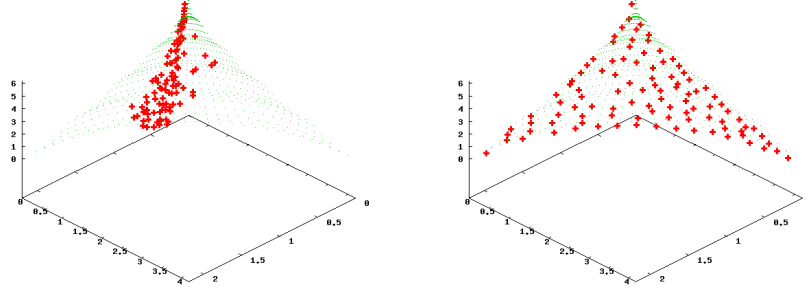


Figure 26: Pareto front after 200 000 function evaluations, problem WFG1, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

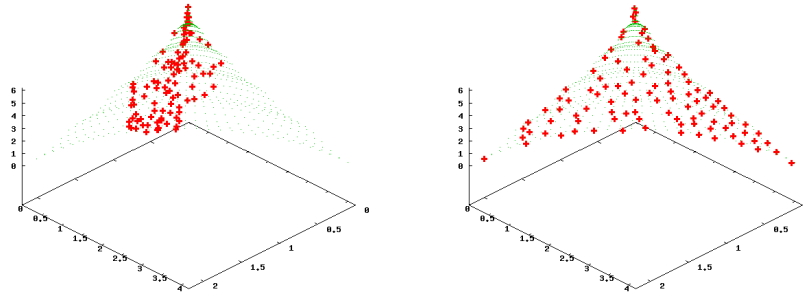


Figure 27: Pareto front after 1 000 000 function evaluations, problem WFG1, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).

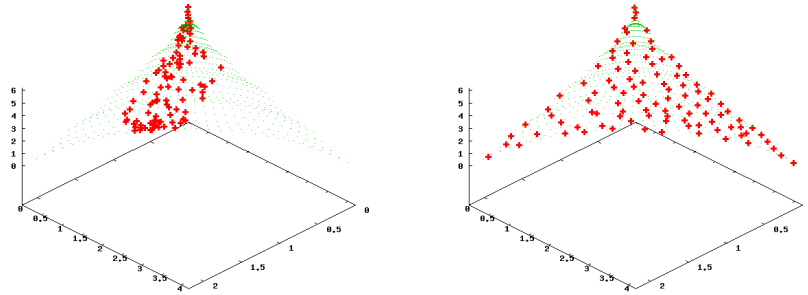


Figure 28: Pareto front after 2 000 000 function evaluations, problem WFG1, μ ARMOGA with population size 4 with crowding distance (left), and with the new proposed algorithm (right).